CLAIMS

1. (Previously presented) A method comprising:
establishing secured communication between a client device and a server device, wherein communication is secured using, at least in part, a plurality of synchronized security sequence values for authentication of secure communications;
storing a security sequence value from the plurality of synchronized security sequence values as a first resynchronization value;
detecting at least one event desynchronizing said secured communication;
requesting resynchronization of security sequence values, requesting resynchronization comprising sending at least a representation of said first resynchronization value from said client device to said server device;
receiving a second resynchronization value in a response to the first resynchronization value; and
reestablishing secured communication using the first resynchronization value and the second resynchronization value.

2. (Previously presented) The method of claim 1, further comprising performing anti-replay filtering using said plurality of synchronized security sequence values.

3. (Previously presented) The method of claim 1, wherein sending at least a representation of said first resynchronization value includes embedding said first resynchronization value in a header of a data packet, a payload of a data packet, or both.

4. (Previously presented) The method of claim 1, further comprising periodically refreshing the stored first resynchronization value with a new value at a selected interval from security sequence values already used in a secured communication session.

5. (Previously presented) A method comprising:
establishing secured communication between a client device and server device; wherein communication is secured using, at least in part, a plurality of synchronized security sequence values for authentication of secure communications;
receiving a request for resynchronization from the client device, the request including at least a representation of a client resynchronization value, the client resynchronization value being a stored synchronized security value of the plurality of synchronized security sequence values;
acknowledging the client request for resynchronization, acknowledging comprising sending the representation of resynchronization value and at least a representation of a server resynchronization value from said server device to said client device; and
reestablishing secured communication using said client resynchronization value and said server resynchronization value.

6. (Cancelled)

7. (Previously presented) The method of claim 5, wherein sending at least a representation of said client and said server resynchronization values includes embedding said client and said server resynchronization values in at least one header, payload, or both of a data packet that conforms to IPsec (Internet Protocol Security) standards.

8. (Original) The method of claim 5, further comprising performing said method using a state machine in network circuitry.

9. (Cancelled)

10. (Previously presented) The method of claim 5, further comprising performing anti-replay filtering using said plurality of synchronized security sequence values.

11. (Original) The method of claim 5, further comprising reestablishing secured communication during a low-power state.

12. (Previously presented) The method of claim 5, further comprising reestablishing secured communication while said client device lacks an active operating system, lacks an active microprocessor, or both.

13. (Cancelled)

14. (Previously presented) An apparatus, comprising;

(a) a security interface to engage in secured communication with at least one network node, wherein said security interface and said at least one network node use a plurality of synchronized security sequence values at least in part to authenticate said secured communication;

(i) a recorder to store at least one security sequence value;

(ii) a desynchronization detector coupled to said security interface;

(iii) a resynchronization requester to send the stored security sequence value to at least one network node after a desynchronization; and

(iv) a verifier to receive feedback from said at least one network node;

(b) a security agent coupled to said at least one network node, comprising:

(i) a request receiver to recognize said stored security sequence value; and

(ii) an acknowledger to send said feedback from said security agent to said security interface, said feedback comprising

said stored security sequence value and a node security sequence value from said network node.

15. (Previously presented) The apparatus of claim 14, wherein the stored security sequence value and the node security sequence value are embedded in at least one header, at least one payload, or both of a data packet that conforms to one or more IPsec (Internet Protocol Security) standards.

16. (Original) The apparatus of claim 14, wherein said stored security sequence value is periodically refreshed with a value at a selected interval from security sequence values already used in a secured communication session.

17. (Previously presented) A computer network security sequence value resynchronizer, comprising:

(a) a sender having at least access to a nonvolatile random access memory;

(b) said sender to transmit a request for resynchronization of security sequence values for authentication of secure communications, the request including a data packet containing at least in part a stored sender resynchronization value from said nonvolatile random access memory over said computer network; and

(c) an acknowledger connected to said computer network to receive said sender resynchronization value from said sender; said acknowledger returning said sender resynchronization value and an acknowledger resynchronization value to said sender as security assurance, the communications to be resynchronized using the sender resynchronization value and the acknowledger resynchronization value.

18. (Cancelled)

19. (Previously presented) The resynchronizer of claim 17, wherein at least one sender and at least one acknowledger are installed on any combination of a server and a client device in a network.

20. (Previously presented) A method comprising:
establishing secured communication between a security interface and a network node, said security interface to resynchronize security sequence values for authentication of secure communications between said security interface and said network node;
storing a first resynchronization value selected by said security interface; and
resynchronizing said security sequence values after a break in said secured communication, said resynchronizing comprising:
sending said first resynchronization value from said security interface to said network node;
sending said first resynchronization value and a second resynchronization value from said network node to said security interface; and
reestablishing said secured communication using said first resynchronization value and said second resynchronization value.

21. (Original) The method of claim 20 further comprising using a security interface as a state machine in network circuitry.

22. (Cancelled)

23. (Original) The method of claim 20 further comprising storing said first resynchronization value in a nonvolatile storage medium.

24. (Previously presented) The method of claim 20 further comprising establishing secured communication using IPsec (Internet Protocol Security) standards.

25. (Previously presented) The method of claim 20, wherein reestablishing the secured communication comprises resynchronizing said secured communication using said first resynchronization value to resynchronize secured data sent from said security interface and using said second resynchronization value to resynchronize secured data sent from said network node.

26. (Previously presented) The method of claim 20 further comprising resynchronizing the secured communication during a low-power state.

27. (Previously presented) The method of claim 20 further comprising resynchronizing the secured communication while said network node lacks an active operating system and/or lacks an active microprocessor.

28. (Previously presented) A method, comprising:
establishing a secured communication between a server device and a client device, said secured communication using a plurality of server security sequence values for authentication of secure communications synchronized with a plurality of client security sequence values;
storing at least one client security sequence value in nonvolatile memory as a saved client security sequence value; and
resynchronizing server and client security sequence values after a desynchronization event, resynchronizing including sending said saved client security sequence value from said nonvolatile memory to said server device and returning said saved client security sequence value from said server device to said client device in a data packet with a server security sequence value.

29. (Cancelled)

30. (Previously presented) The method of claim 28, said storing further comprising periodically refreshing said saved client security sequence value with a later security sequence value.

31. (Previously presented) A machine-readable medium having stored thereon data representing sequences of instructions that, when executed by a processor, cause the processor to perform operations comprising:

establishing a secured communication between a client device and server device; wherein communication is secured using at least in part a plurality of synchronized security sequence values for authentication of secure communications;

storing a security sequence value of the plurality of synchronized security sequence values as a first resynchronization value;

detecting desynchronization of the secured communication; and

requesting resynchronization of security sequence values, wherein requesting resynchronization includes sending the first resynchronization value from the client device to the server device;

receiving a second resynchronization value in response to the first resynchronization value; and

reestablishing secured communication using the first resynchronization value and the second resynchronization value.

32. (Previously presented) The medium of claim 31, further comprising instructions that, when executed by the processor, cause the processor to perform operations comprising:

periodically refreshing the stored first resynchronization value with a new value from security sequence values already used in a secured communication session.

33. (Previously presented) A machine-readable medium having stored thereon data representing sequences of instructions that, when executed by a processor, cause the processor to perform operations comprising:

establishing secured communication between a client device and server device; wherein communication is secured using, at least in part, a plurality of synchronized security sequence values for authentication of secure communications;

receiving a request for resynchronization from the client device, the request including a resynchronization value, the resynchronization value being a stored synchronized security sequence of the plurality of security sequence values;

acknowledging the client request for resynchronization, acknowledging comprising sending resynchronization value and a server resynchronization value from the server device to the client device; and

reestablishing secured communication using the client resynchronization value and the server resynchronization value.

34. (Previously presented) The medium of claim 33, further comprising instructions that, when executed by the processor, cause the processor to perform operations comprising reestablishing secured communication during a low-power state.

35. (Previously presented) The medium of claim 33, further comprising reestablishing secured communication while the client device lacks an active operating system, lacks an active microprocessor, or both.

36. (Previously presented) The method of claim 1, wherein the response further includes the first resynchronization value.

37. (Previously presented) The method of claim 36, wherein the first resynchronization value is contained in payload data of the response.

38-39. (Cancelled)